Localization in Sensor Networks I

Jie Gao
Computer Science Department
Stony Brook University

STONY BROOK
STATE UNIVERSITY OF NEW YORK

Find where the sensor is…

- Location information is important.
  1. Devices need to know where they are.
     - Sensor tasking: turn on the sensor near the window…
  2. We want to know where the data is about.
     - A sensor reading is too hot – where?
  3. It helps infrastructure establishment.
     – geographical routing
     – sensor coverage.

GPS is not always good

- Requires clear sky, doesn't work indoor.
- Too expensive.
  - A $1 sensor with a $100 GPS?

Localization algorithm:

- (optional) Some nodes (anchors or beacons) know their locations (e.g., through GPS).
- Nodes make local measurements;
  - Distances or angles between two neighbors.
- Communicate between each other;
- Infer location information from these measurements.

Localization problem

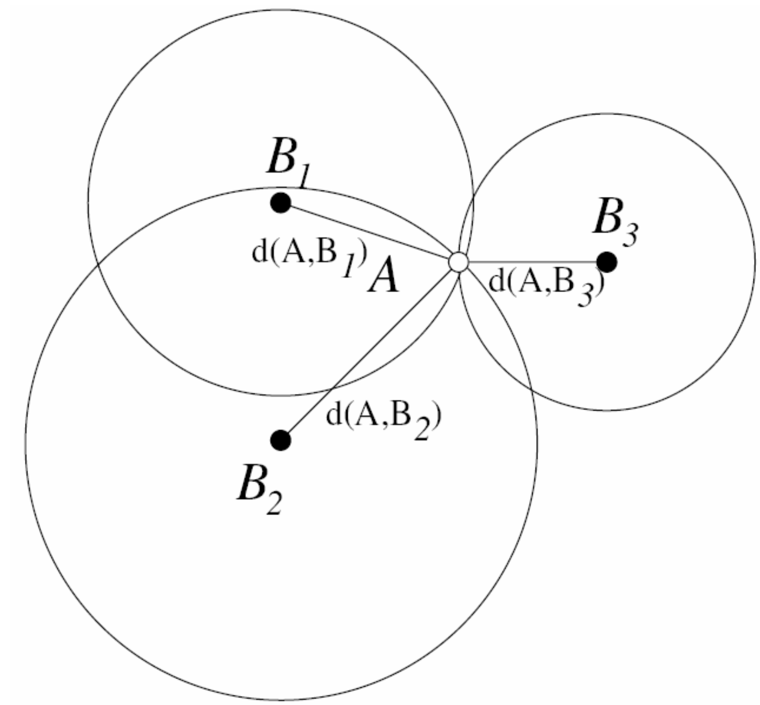
- Output: nodes' location.
  - Global location, e.g., what GPS gives.
  - Relative location.
- Input:
  - Connectivity, hop count (under UDG model).
    - Nodes with k hops away are within Euclidean distance k.
    - Nodes without a link must be at least distance 1 away.
  - Distance measurement of an incoming link.
  - Angle measurement of an incoming link.
  - Combinations of the above.

Distance Measurements

- Received Signal Strength Indicator (RSSI)
  - The further away, the weaker the received signal.
  - Mainly used for RF signals.
- Time of Arrival (ToA) or Time Difference of Arrival (TDoA)
  - Signal propagation time translates to distance.
  - RF, acoustic, infrared and ultrasound.

Time of Arrival (ToA)

- Used in GPS.
- Triangulation.
- Need synchronization.
- If round-trip time is used, then accurate clocks are only needed at the anchors.

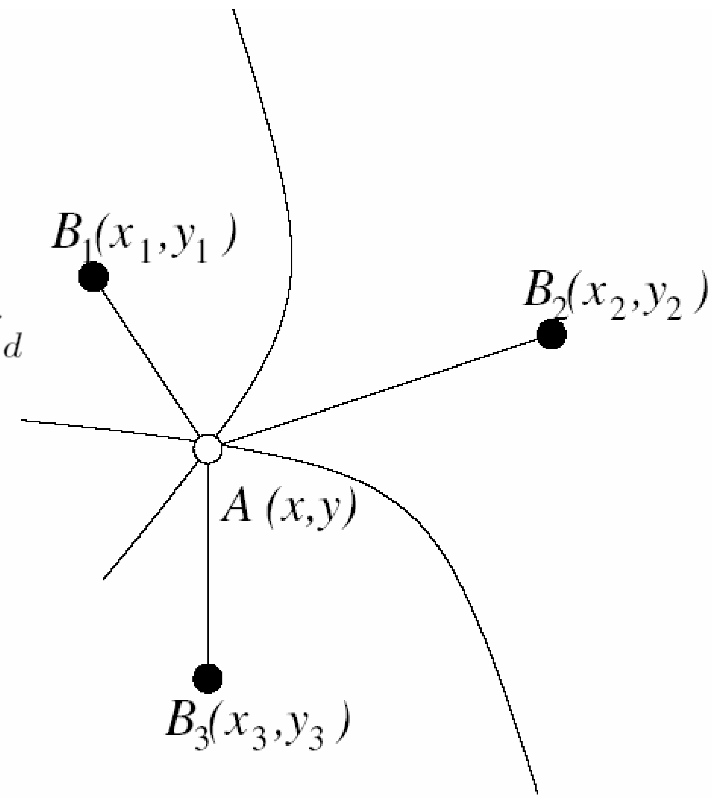


Time Difference of Arrival (TDoA)

- Anchor B1 and B2 send signal to A simultaneously. The time difference of arrival is recorded.
- A stays on the hyperbola:

$$\sqrt{(x-x_1)^2+(y-y_1)^2}-\sqrt{(x-x_2)^2+(y-y_2)^2}=\delta_d$$

- Do this for B2 and B3.
- A stays at the intersection of the two hyperbolas.
- If the two hyperbolas have 2 intersections, one more measurement is needed.

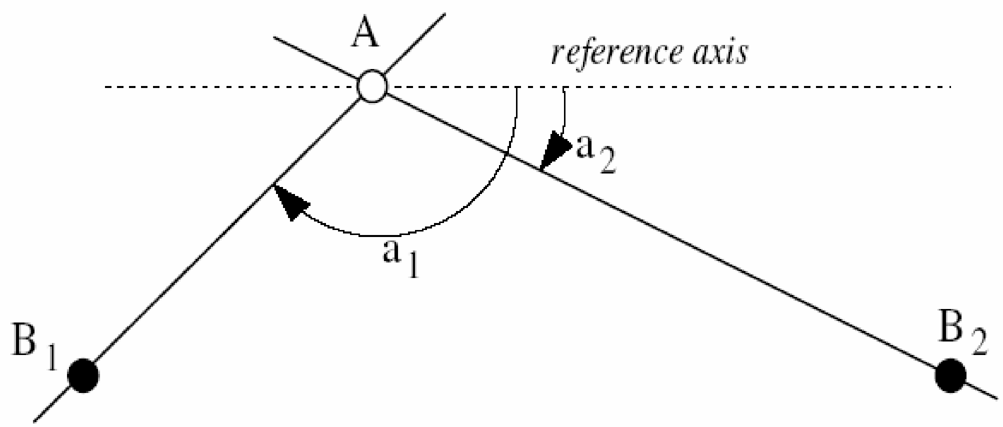


Angle Measurements

- Angle of Arrival (AoA)
  - Determining the direction of propagation of a radio-frequency wave incident on an antenna array.
- Directional Antenna
- Special hardware, e.g., laser transmitter and receivers.

Angle of Arrival (AoA)

- A measures the direction of an incoming link by radio array.
- By using 2 anchors, A can determine its position.



Localization algorithms

- **Anchor-based**
  - Some nodes know their locations, either by a GPS or as pre-specified.
- **Anchor-free**
  - Relative location only.
  - A harder problem, need to solve the global structure. Nowhere to start.
- **Range-based**
  - Use range information (distance estimation).
- **Range-free**
  - No distance estimation, use connectivity information such as hop count.

Required Papers

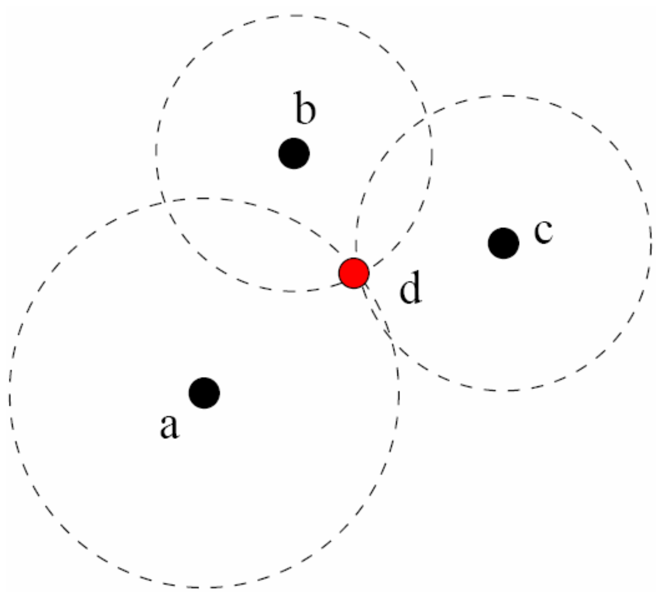
- **[Savvides01]** A. Savvides, C.-C. Han, and M. B. Strivastava. **Dynamic fine-grained localization in ad-hoc networks of sensors**. Proc. MobiCom 2001.

- **[Eren04]** Tolga Eren, David Goldenberg, Walter Whitley, Yang Richard Yang, A. Stephen Morse, Brian D.O. Anderson and Peter N. Belhumeur, **Rigidity, Computation, and Randomization of Network Localization**. In Proceedings of IEEE INFOCOM, Hong Kong, China, April 2004.

Multilateration: use plane geometry

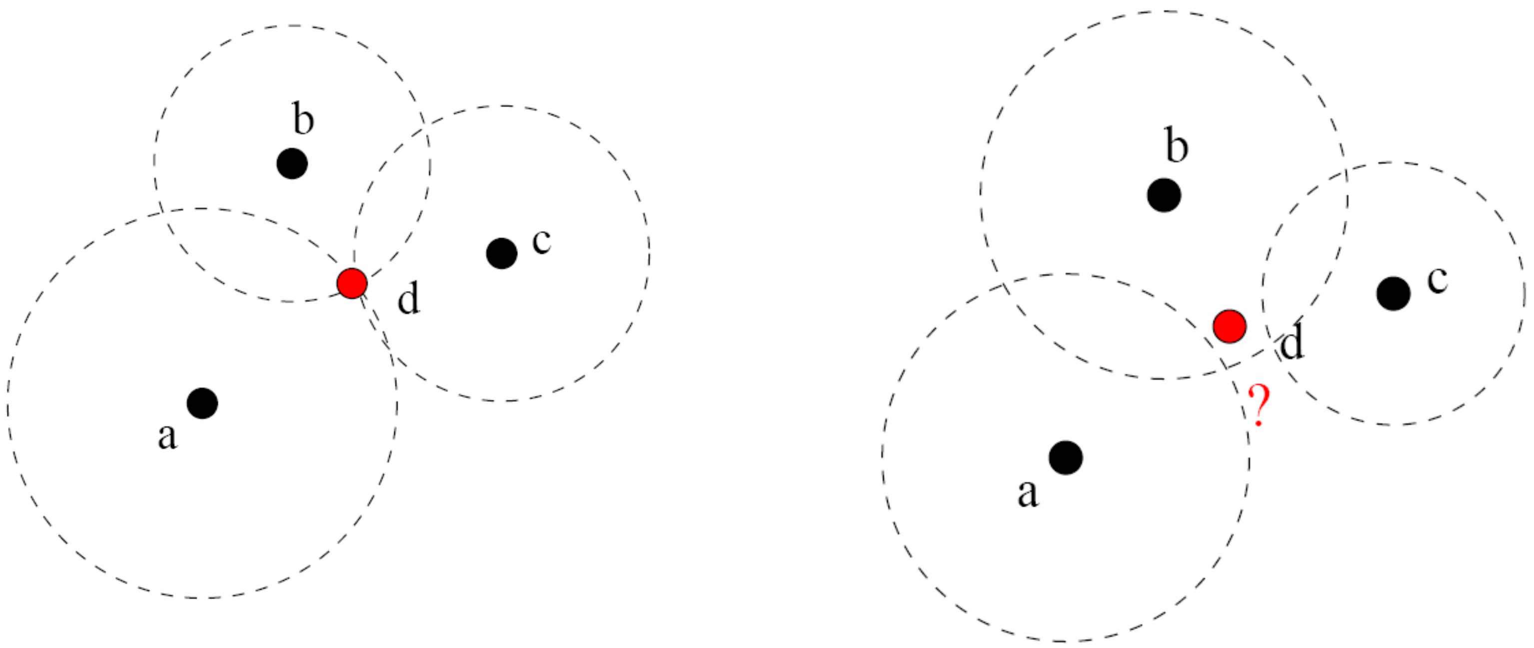
Triangulation, trilateration

- Anchors advertise their coordinates & transmit a reference signal
- Other nodes use the reference signal to estimate distances anchor nodes.



Triangulation, trilateration

- Distance measurements are noisy!
- Solve an optimization problem: minimize the mean square error.



Problem Formulation

- k beacons at positions (x_i, y_i)
- Assume node 0 has position (x_0, y_0)
- Distance measurement between node 0 and beacon i is r_i
- Error:

$$f_i = r_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

- The objective function is

$$F(x_0, y_0) = \min \sum f_i^2$$

- This is a non-linear optimization problem

Linearization and Min Mean Square Estimate

- Ideally, we would like the error to be 0

$$f_i = r_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} = 0$$

- Re-arrange:

$$(x_0{}^2 + y_0{}^2) + x_0(-2x_i) + y_0(-2y_i) - r_i{}^2 = -x_i^2 - y_i^2$$

- Subtract the last equation from the previous ones to get rid of quadratic terms.

$$2x_0(x_k - x_i) + 2y_0(y_k - y_i) = r_i{}^2 - r_k{}^2 - x_i^2 - y_i^2 + x_k^2 + y_k^2$$

- Note that this is linear.

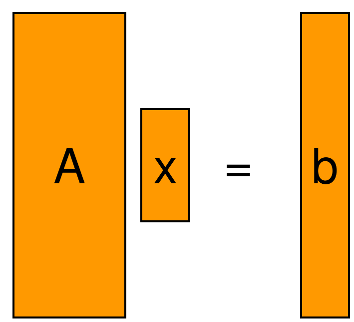
Linearization and Min Mean Square Estimate

- In general, we have an over-constrained linear system

$$Ax = b$$

$$b = \begin{bmatrix} r_1^2 - r_k^2 - x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ r_2^2 - r_k^2 - x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 - x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix} \qquad A = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) \\ 2(x_k - x_2) & 2(y_k - y_2) \\ \vdots & \vdots \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) \end{bmatrix}$$

$$x = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$



Solve using the Least Square Equation

The linearized equations in matrix form become

$$Ax = b$$

Now we can use the least squares equation to compute an estimation.

$$x = (A^T A)^{-1} A^T b$$

How to solve it in an embedded system?

- Check conditions
  - Beacon nodes must not lie on the same line

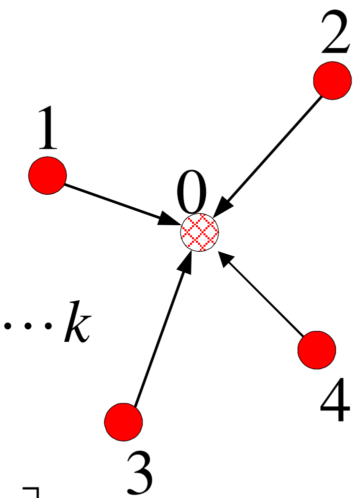
- For ToA, TDoA, if we use acoustic signals, how to solve for the speed of sound?

Acoustic case: Also solve for the speed of sound

With at least 4 beacons,

$$f_i = s t_{i0} - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

Speed of sound (s) — Time measurement (t_{i0})



$$i = 1, 2 \cdots k$$

This can be linearized to the form

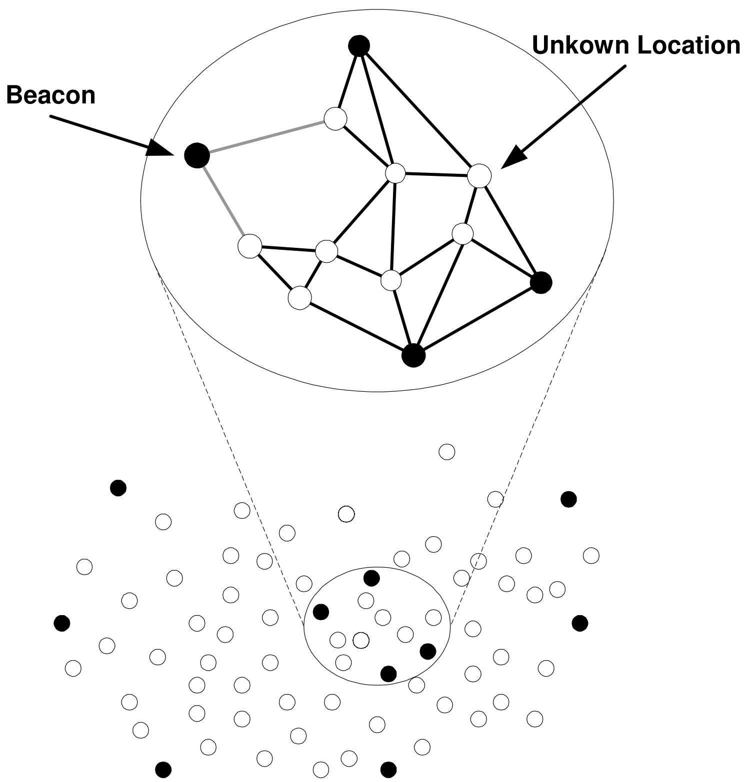
where

$$Ax = b$$

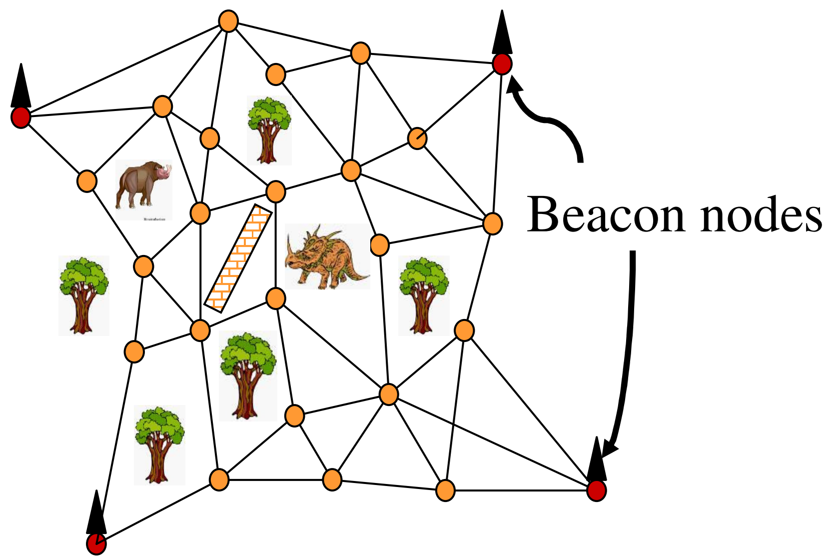
$$b = \begin{bmatrix} -x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ -x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \vdots \\ -x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix} \quad A = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) & t_{k0}^2 - t_{10}^2 \\ 2(x_k - x_2) & 2(y_k - y_2) & t_{k0}^2 - t_{20}^2 \\ \vdots & \vdots & \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) & t_{k0}^2 - t_{(k-1)0}^2 \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ y_0 \\ s^2 \end{bmatrix}$$

$$x = (A^T A)^{-1} A^T b$$

The Node Localization Problem



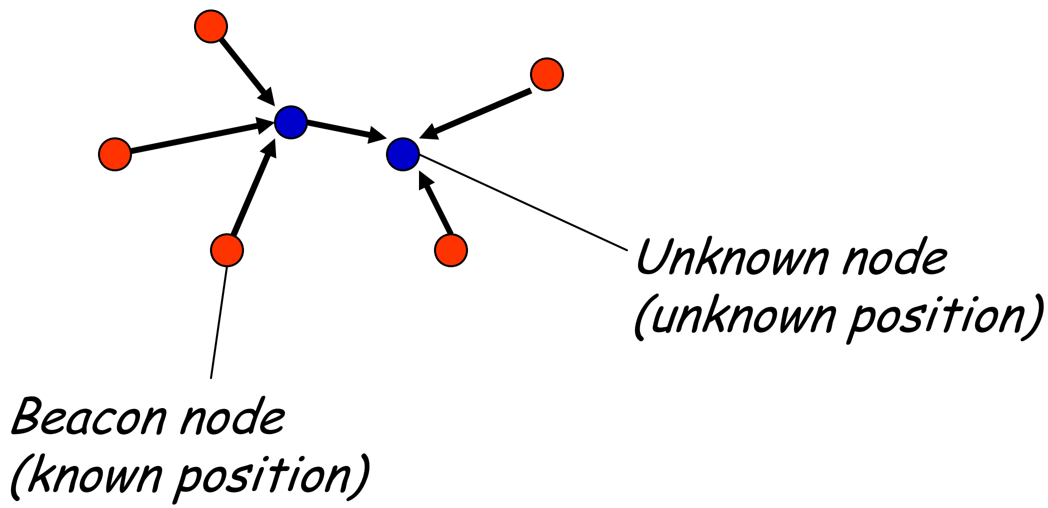
Randomly Deployed Sensor Network



- Localize nodes in an ad-hoc multihop network
- Based on a set of inter-node distance measurements

Iterative multilateration

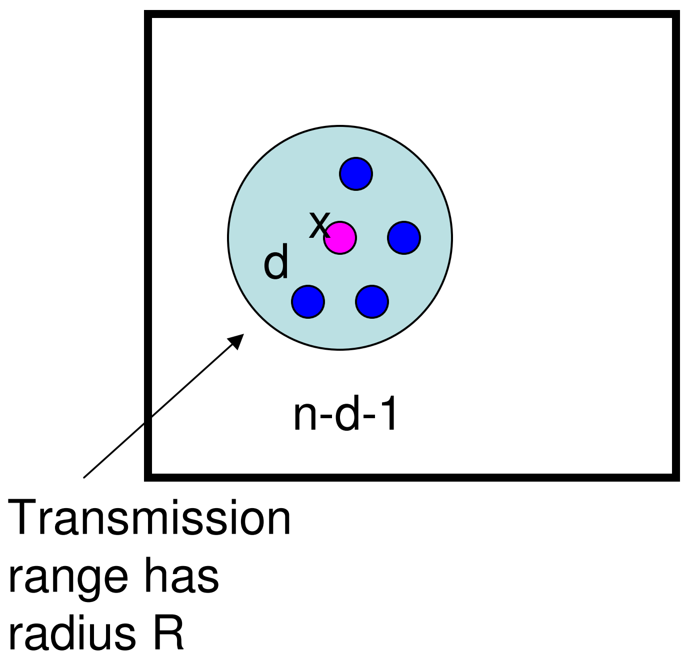
- Iterative multilateration
  - a node with at least 3 neighboring beacons estimates its position and becomes a beacon.
  - Iterate until all nodes with 3 beacons are localized.



Connectivity matters! Each node needs at least 3 neighbors.

Iterative multilateration: how many beacons?

- n nodes deployed randomly in a square of side L,
- P(d)=Pr{a node x has degree d}=?



Probability that one node falls inside the transmission range of x?

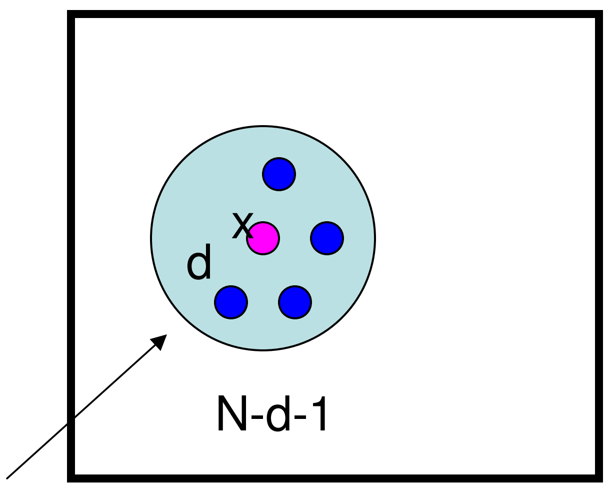
$$p = \frac{\pi R^2}{L^2}$$

Binomial distribution

$$P(d) = p^d \cdot (1-p)^{n-d-1} \cdot \binom{n-1}{d}$$

Iterative multilateration: how many beacons?

- When n tends to infinity, the binomial distribution converges to a Poisson distribution.



Transmission range has radius R

Probability that one node falls inside the transmission range of x?

$$p = \frac{\pi R^2}{L^2} \qquad \lambda = n \cdot p$$

Binomial distribution

↓

Poisson distribution

$$P(d) = \frac{\lambda^d}{d!} \cdot e^{-\lambda}$$

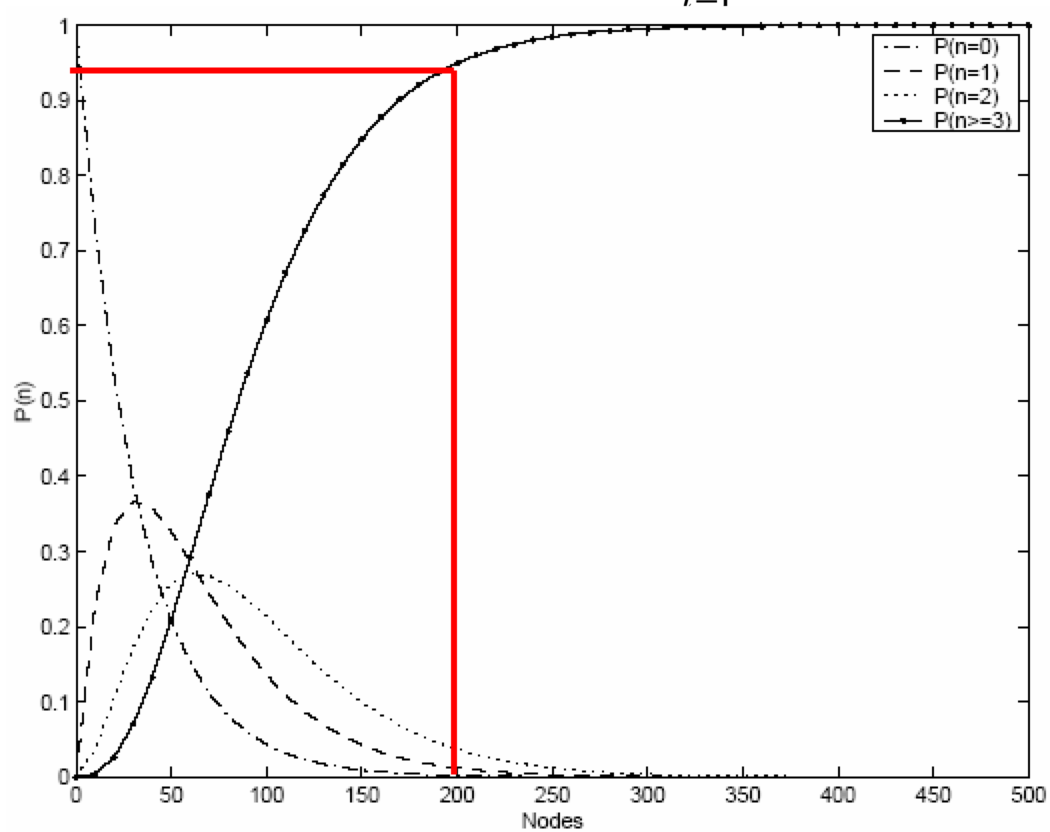
Iterative multilateration: how many beacons?

$$P(d) = \frac{\lambda^d}{d!} \cdot e^{-\lambda} \qquad P(\geq d) = 1 - \sum_{i=1}^{n-1} P(i)$$

100 by 100 field
Sensor range:10

Probability of a node with 0, 1, 2, ≥ 3 neighbors.

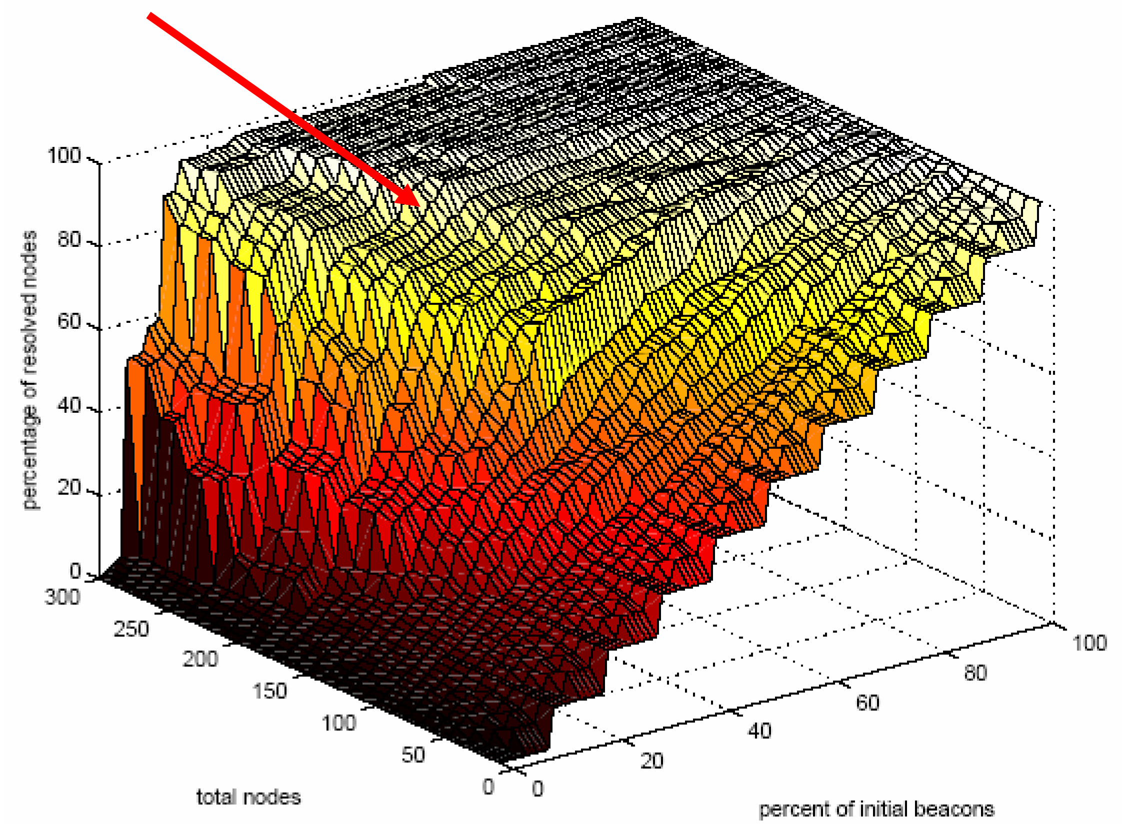
With 200 nodes, P(≥ 3) is about 95%.



Iterative multilateration: how many beacons?

With 200 nodes, P(≥ 3) is about 95%.

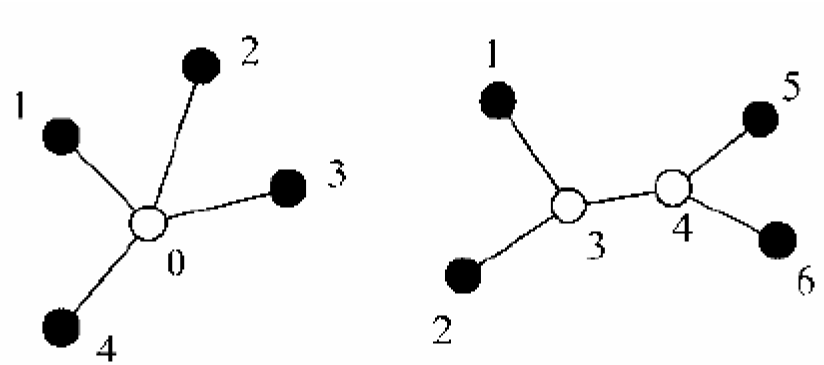
With 200 nodes, we need about 50~60 beacons to localize about 90% of the nodes. That's ¼ of the total number of nodes.



Problems of iterative multilateration

Problems

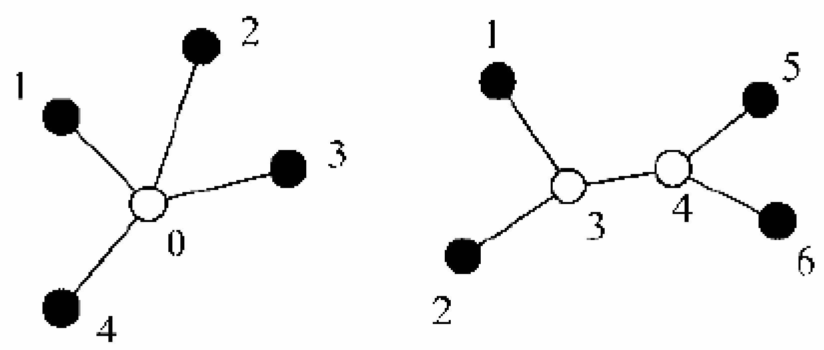
1. Requires a large fraction of beacons.
2. Error accumulates.
3. It gets stuck --- not all nodes with 3 or more neighbors can be resolved.



Problems of iterative Multilateration

Problems

1. Requires a large fraction of beacons.
2. Error accumulates. ← Mass-spring optimization.
3. It gets stuck --- not all nodes with 3 or more neighbors can be located. ← Global optimization (to be discussed next class)

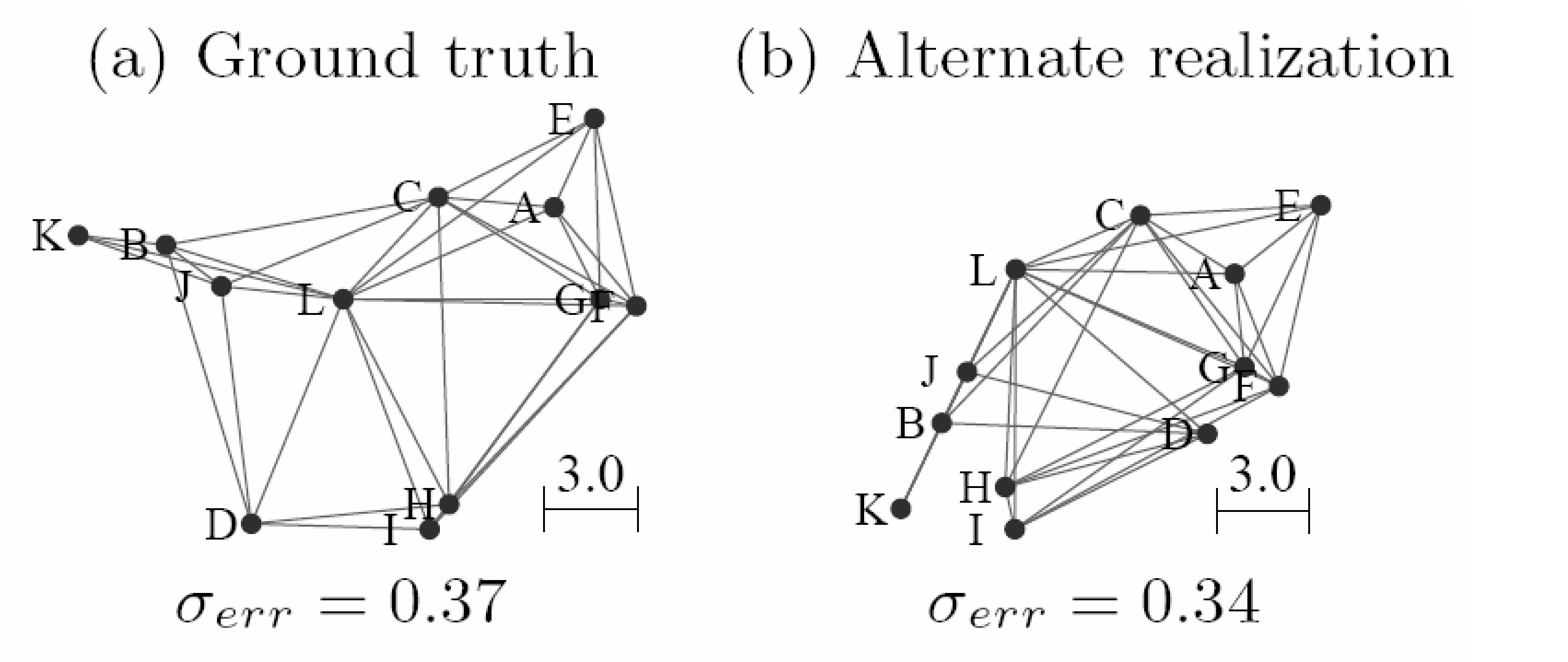


However, optimization does not solve:

Ambiguity in localization

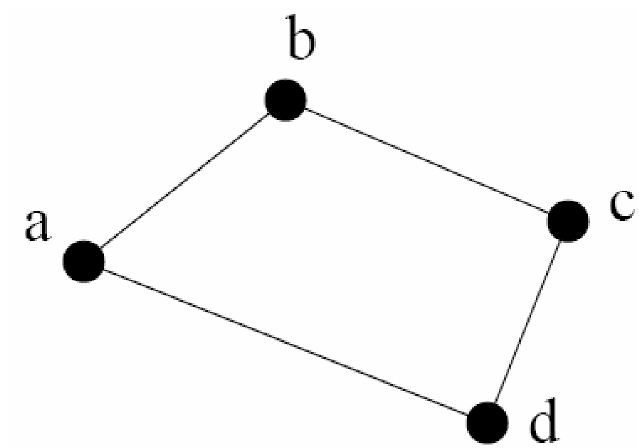
Ambiguity in localization

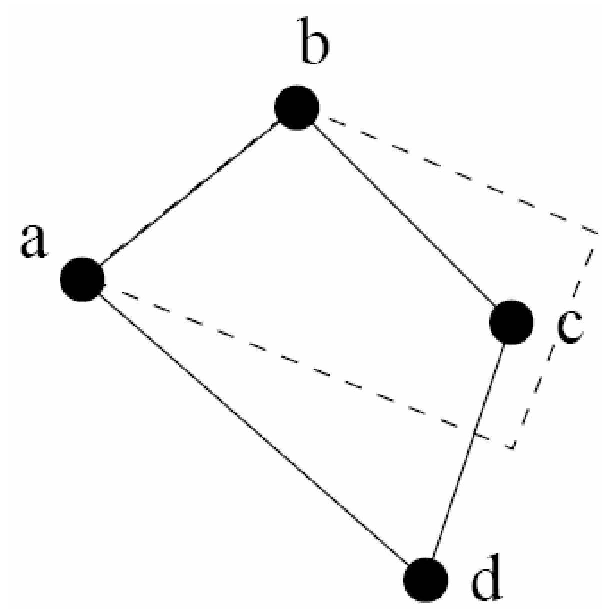
- Same distances, different realization.



Continuous deformation

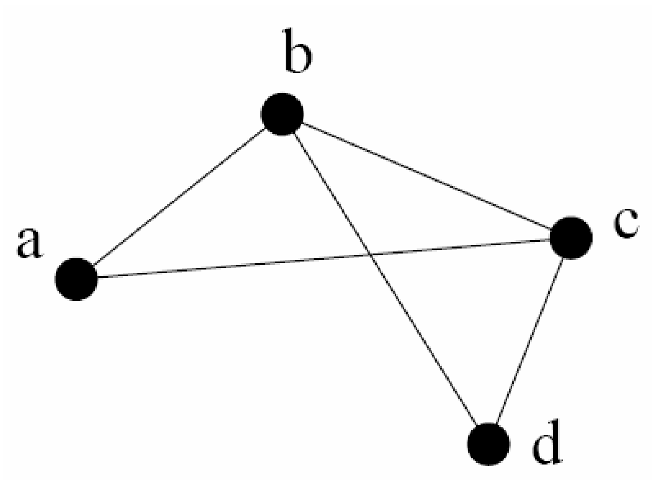
- Nodes move continuously without violating the distance constraints.

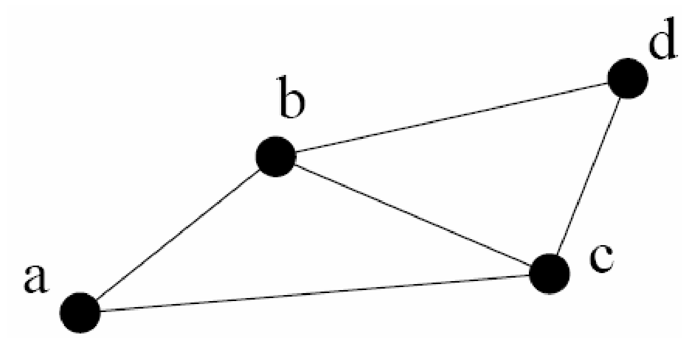




Flip

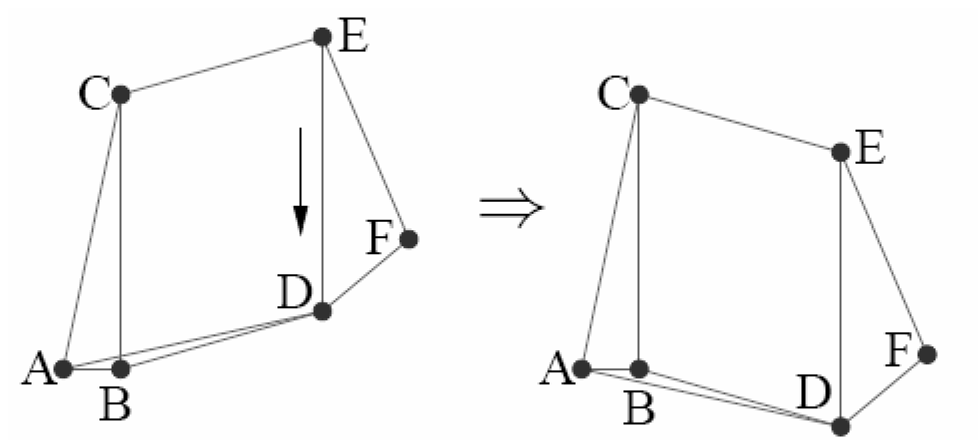
- No continuous deformation, but subjects to global flipping.





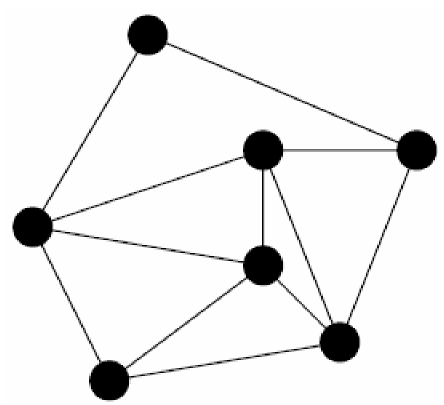
Discontinuous flex ambiguity

- Remove AD, flip ABD up, insert AD.
- No continuous deformation in between.
- But both are valid realization of the distances.



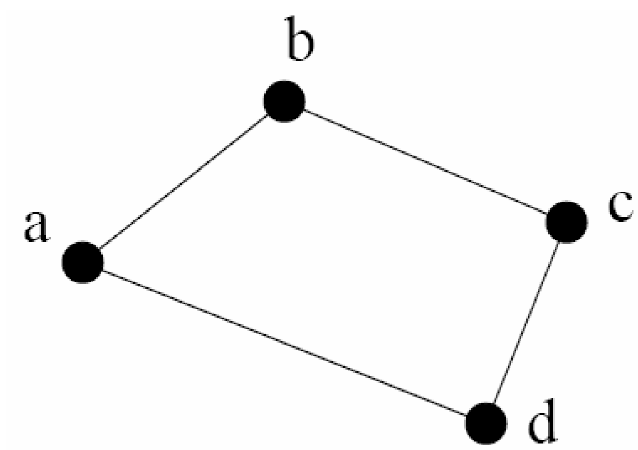
Rigidity theory

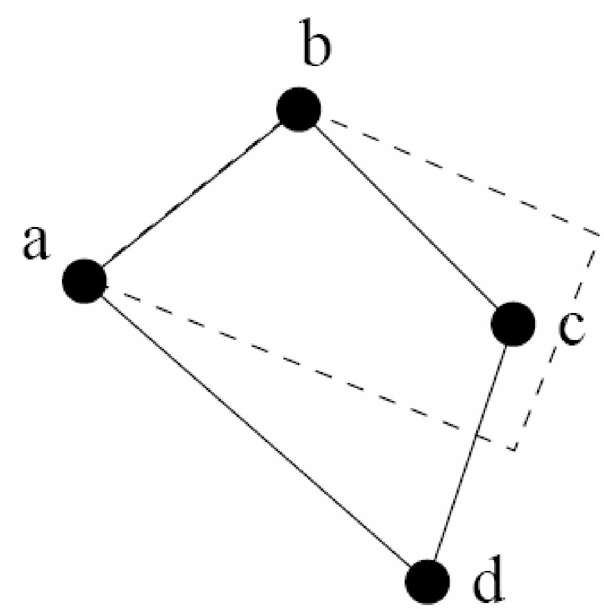
Given a system of rigid bars and hinges in 2D, does it have a continuous deformation? Multiple realizations?



Rigidity theory

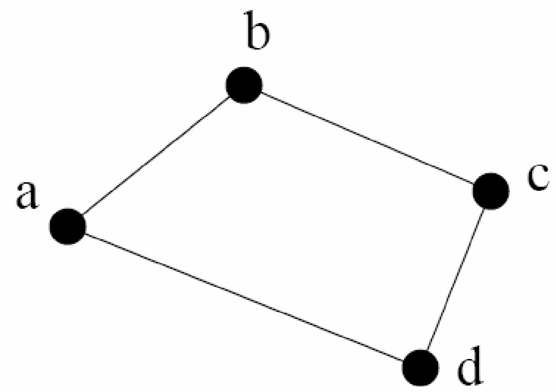
- Given a set of rigid bars connected by hinges, rigidity theory studies whether you can move them continuously.



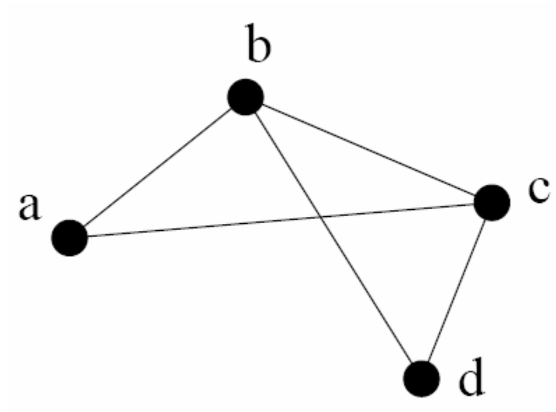


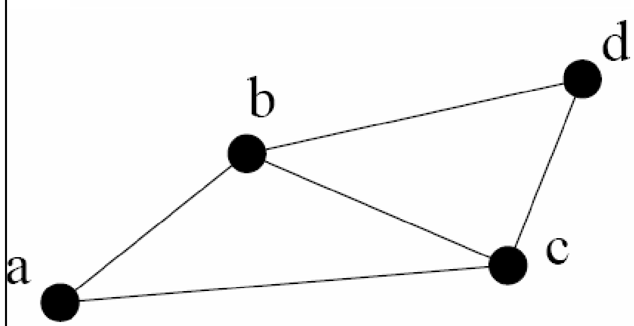
Rigidity and global rigidity

Not rigid

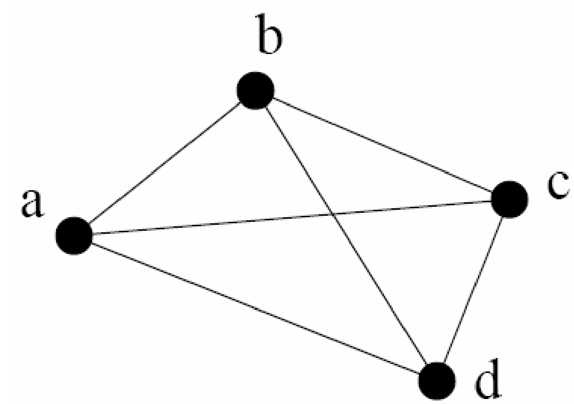


Rigid=
No continuous
deformation





Globally rigid=
unique realization



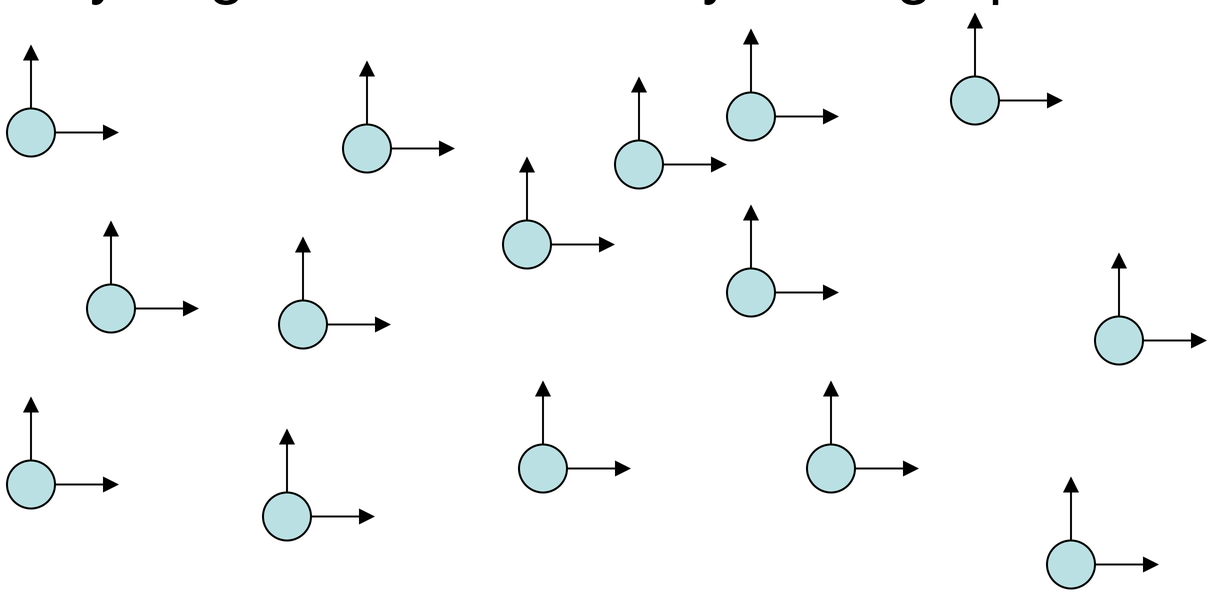
What we want!

Intuition on rigidity (not global rigidity yet)

How many distance constraints are necessary to limit a framework to only trivial motion?

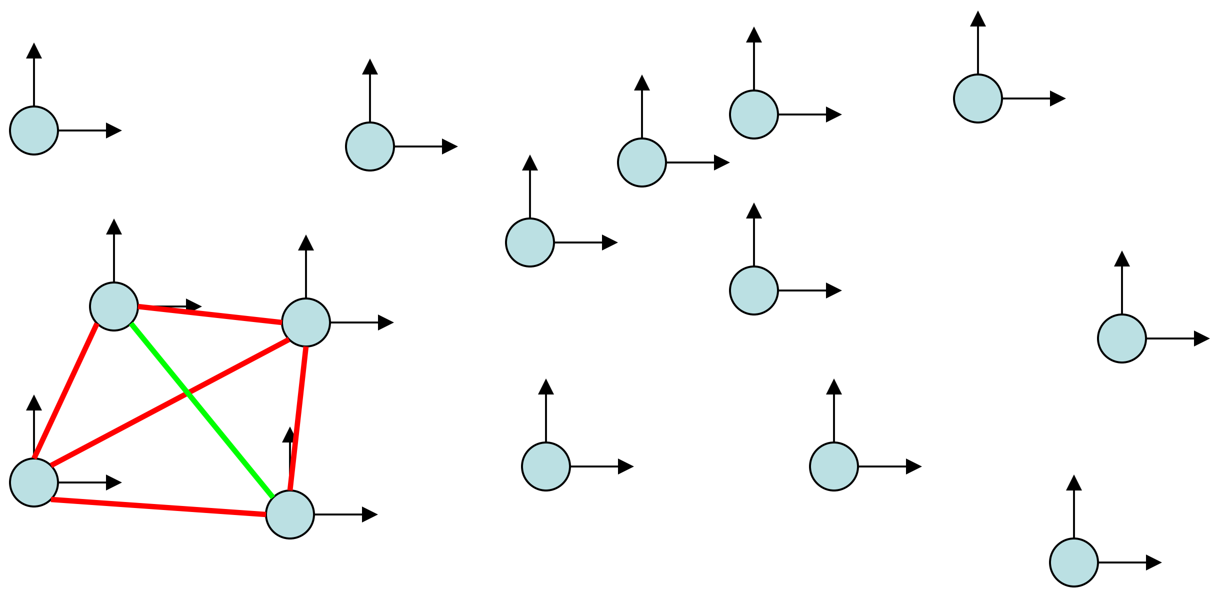
==

How many edges are necessary for a graph to be rigid?



Total degrees of freedom: 2n

How many edges are necessary to make a graph of *n* nodes rigid?

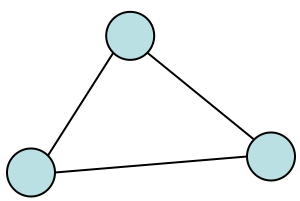


Each edge can remove a single degree of freedom

Rotations and translations will always be possible, so at least ***2n-3*** edges are necessary for a graph to be rigid.

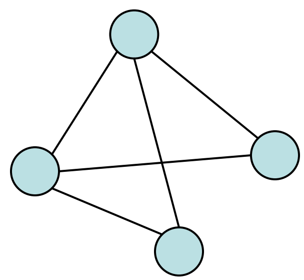
Are 2n-3 edges sufficient?

$n = 3,\ 2n\text{-}3 = 3$



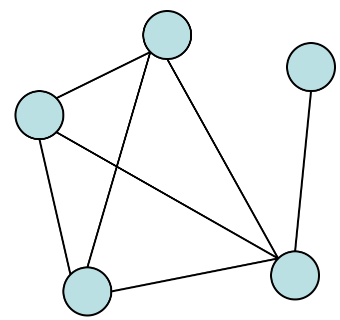
yes

$n = 4,\ 2n\text{-}3 = 5$



yes

$n = 5,\ 2n\text{-}3 = 7$



no

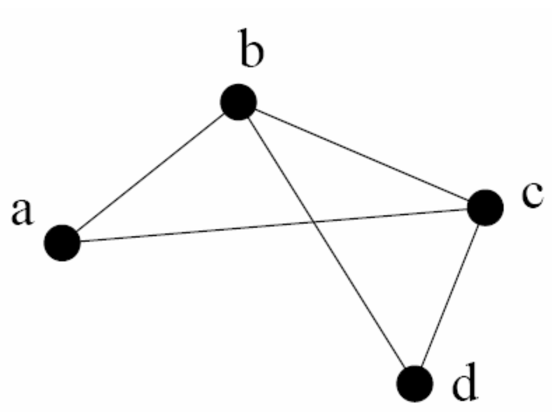
Further intuition

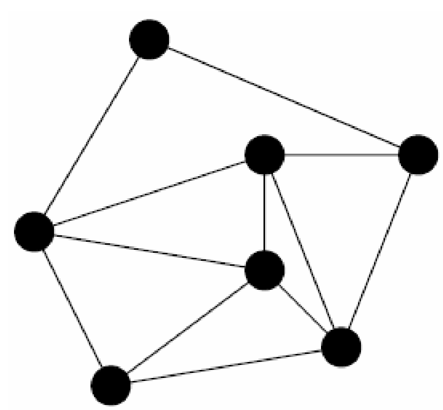
- Need at least *2n-3* “well-distributed” edges.
- If a subgraph has more edges than necessary, some edges are redundant*.*
- Non-redundant edges are independent, i.e., they remove a degree of freedom each.
- Therefore, *2n-3* independent edges guarantee rigidity.

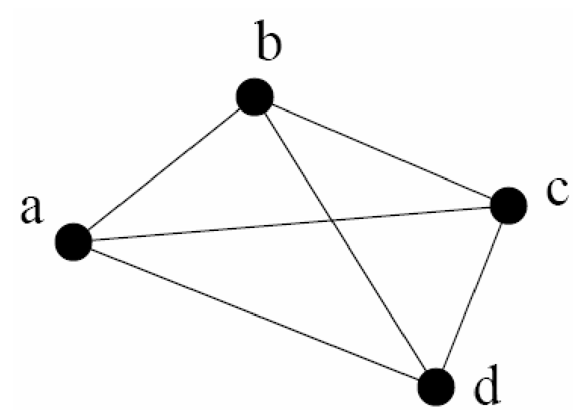
Laman condition

Laman graph: it has 2n-3 edges and no subgraph of k vertices has more than *2k-3* edges.

Laman condition: A graph is rigid if it contains a Laman graph.







How does a Laman graph look like?

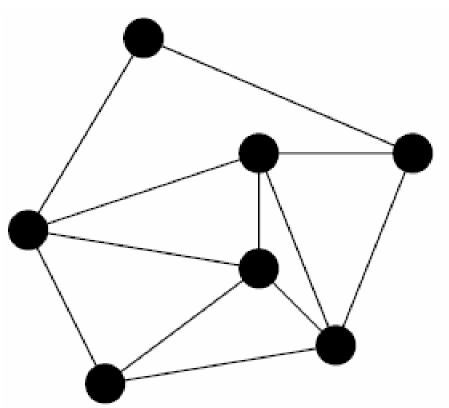
Henneberg constructions

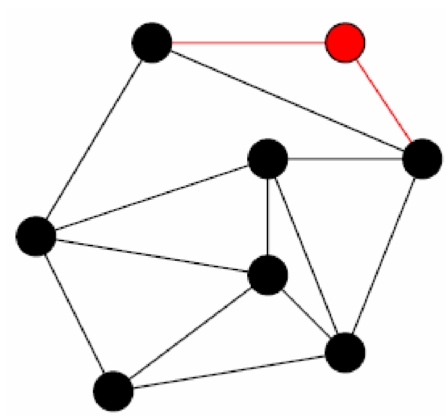
- **Henneberg constructions** (Tay-Whiteley): inductive, add one vertex at a time:

- Start with an edge. At each step, add a new vertex
  - Type I step: join the vertex to two old vertices via two edges
  - Type II step: join the vertex to three old vertices with at least one edge in between, via three edges. Remove an old edge between the three endpoints.

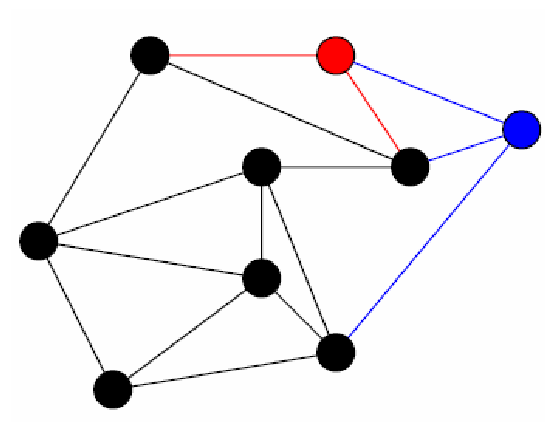
Henneberg constructions

- Type I step: join the vertex to two old vertices via two edges
- Type II step: join the vertex to three old vertices with at least one edge in between, via three edges. Remove an old edge between the three endpoints.





Type I



Type II

Laman = Henneberg construction

- A graph constructed by Henneberg construction is Laman.
- Every Laman graph can be constructed by using Henneberg construction.

Henneberg ➔ Laman

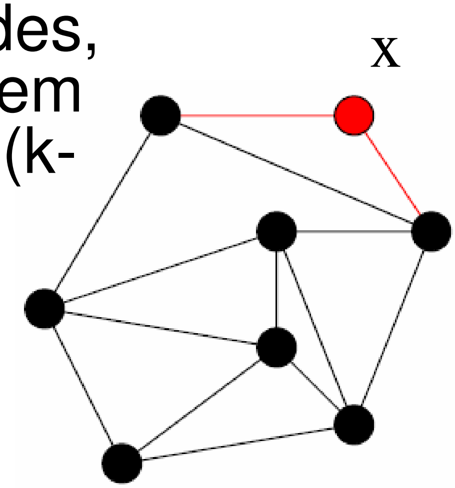
Claim: A graph constructed "Henneberg-ly" is Laman.

Proof: By induction. Suppose the current graph G is Laman with n vertices, 2n-3 edges.

Type I: Add node x. Now n+1 vertices, and 2n-3+2=2(n+1)-3 edges.

Similarly, for a subgraph with k nodes, if it does not include x, by the induction hypothesis, there are $\leqslant$ 2k-3 edges.

If the subgraph includes x, for the other k-1 nodes, there are at most 2(k-1)-3 edges between them (induction hypothesis), in total there are $\leqslant$ 2(k-1)-3 +2 = 2k-3 edges



Henneberg ➔ Laman

Type II: Add node x. We have n+1 vertices, and 2n-3+3-1=2(n+1)-3 edges.

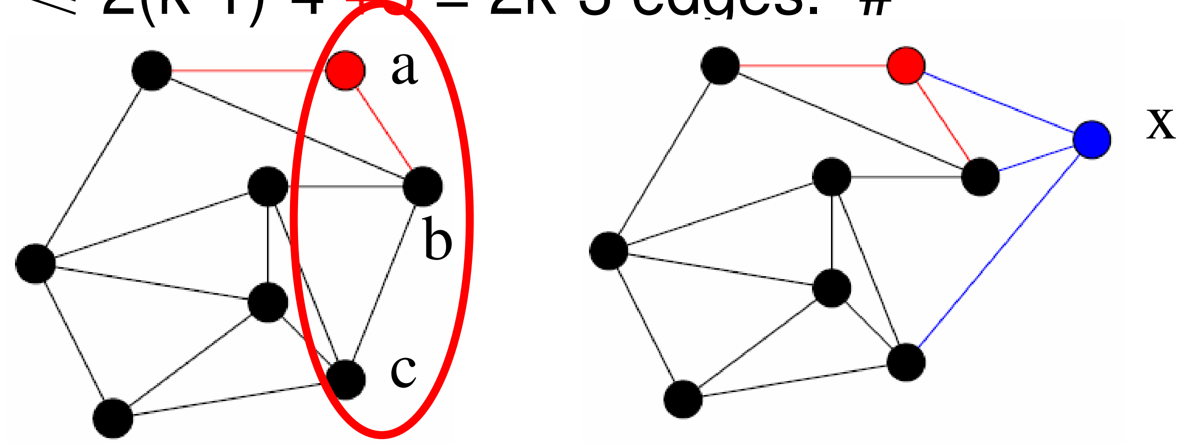
For a subgraph with k nodes, if it does not include x, by the induction hypothesis, there are ⩽ 2k-3 edges.

If the subgraph includes x, for the other k-1 nodes, there are at most

1. 2(k-1)-3 edges, if not all of a, b, c are included.
2. 2(k-1)-4 edges, if a,b,c are all included.

Add x, for case 1, there are ⩽ 2(k-1)-3 +2 = 2k-3 edges.

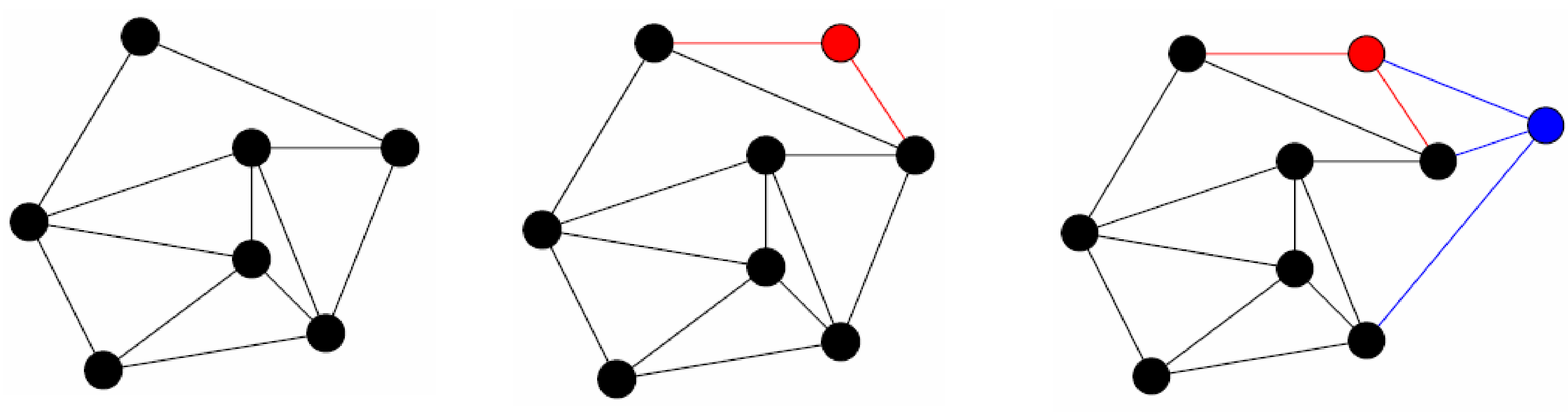
For case 2, there are ⩽ 2(k-1)-4 +3 = 2k-3 edges. #



Laman ➔ Henneberg

Claim: Each Laman graph has a Henneberg construction.

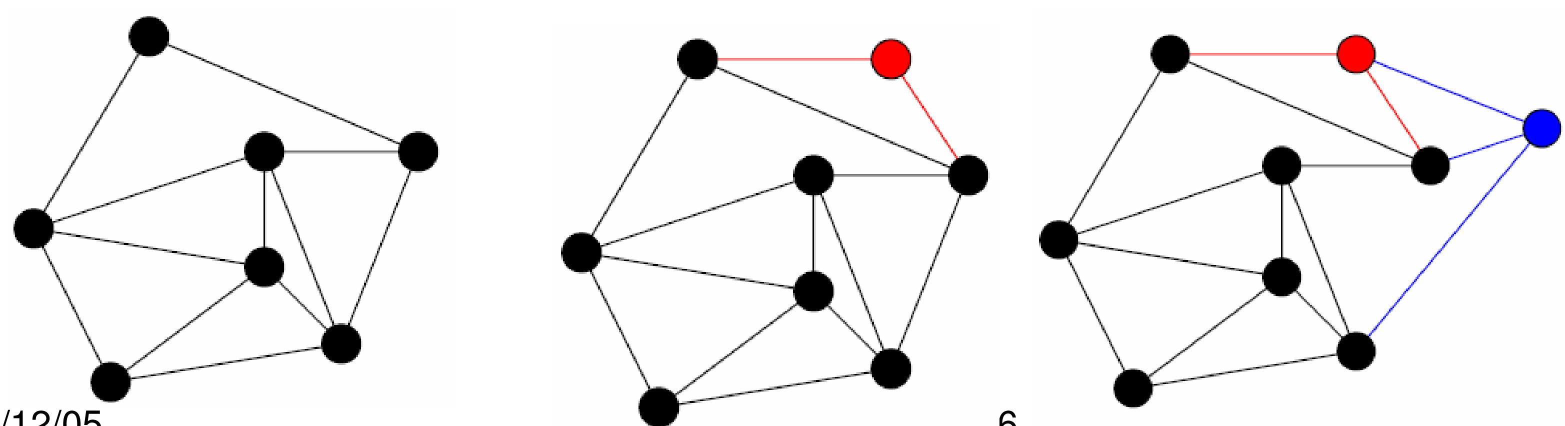
- If m=2n-3, there exists at least one vertex of degree 2 or 3.
- Otherwise, all nodes have degree 4. Thus we have at least 4n/2=2n edges. ➔ contradiction.



Laman ➔ Henneberg

Claim: Each Laman graph has a Henneberg construction.

- If degree 2: remove the vertex and its adjacent edges (Type I step in reverse)
- If degree 3: remove the vertex and the edges to its three neighbors {a, b, c}. They can't span all three edges (else violate 2k-3 for k=4, e.g., {a, b, c, x}). Put one edge between them. (Type II step in reverse).
- Laman still holds, so we can continue.



Hennerberg construction implies…

- The subgraph examined by iterative multi-lateration is rigid.
  - Start with three nodes (with known locations).
  - Add 1 new node with 3 edges to existing nodes.
- Such a graph is named "trilateration graph".

Laman theorem in 3D?

Laman condition in 3D?

A graph is generically minimally rigid in 3D if and only if it has *3n-6* edges and no subgraph of k vertices has more than *3k-6* edges.

Unfortunately, the condition is necessary but not sufficient.

It's a long open problem what is the combinatorial condition for rigidity in 3D.

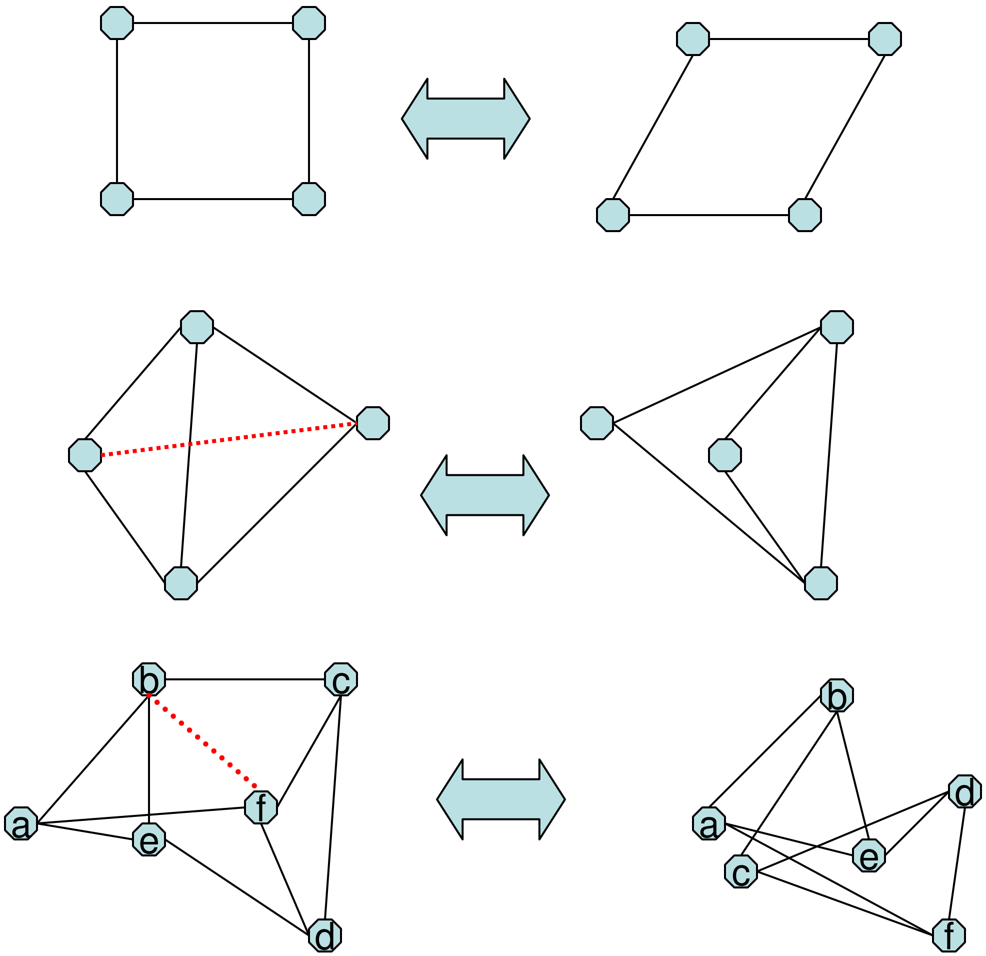
Laman graph in 3D?

Summary of what we know so far

- 2D Rigidity, Laman graph

- But, rigidity does not mean global rigidity.
- For localization, we really want global rigidity.

Global rigidity



Solution:

G must *rigid*

G must be 3-connected, i.e. Connected after removal of 2 Vertices.

G must be *redundantly rigid:* It must remain rigid upon removal of any single edge

Hennerberg construction implies…

- The subgraph examined by iterative multi-lateration is globally rigid.
  - Start with three nodes (with known locations).
  - Add 1 new node with 3 edges to existing nodes.
- Such a graph is named “trilateration graph”.

Paper presentation and critique 9/19

[Priyantha05] Nissanka B. Priyantha, Hari Balakrishnan, Erik D. Demaine, Seth Teller, **Mobile-Assisted Localization in Wireless Sensor Networks**, INFOCOM'05.

[Li05] Zang Li, Wade Trappe, Yanyong Zhang, Badri Nath, **Robust statistical methods for securing wireless localization in sensor networks**, IPSN'05.